I. Amendments to the Claims

Please amend the claims as follows with the following version of the claims in accordance with revised 37 CFR § 1.121.

B1

1. (Canceled).

2. (Canceled).

3. (Canceled).

4. (Canceled).

5. (Canceled).

6. (Canceled).

7. (Canceled).

8. (Canceled).

9. (Canceled).

10. (Canceled).

B1

11. (Previously Presented) A method for enabling a program written in untrusted code to access a native operating system resource, comprising the steps of:

having a trusted login service listen on a named pipe for login requests;

responsive to a login request, wherein the login request contains an identifier for a uniquely-named response pipe, having the trusted login service request a native operating system identifier;

returning to the program via the uniquely-named response pipe the native operating system identifier, wherein the uniquely-named response pipe and the named pipe are not identical;

in an authentication framework, using the native operating system identifier to create a credential object; and

using the credential object to login to the native operating system to enable the program to access the resource.

12. (Previously Presented) The method as described in claim 11 wherein the native operating system supports named-pipe servers.

13. (Previously Presented) The method as described in claim 12 wherein the program is written in an interpreted language.

B1

14. (Original) The method as described in claim 11 wherein the authentication framework is a pluggable authentication mechanism (PAM) having a set of application programming interfaces (APIs).

15. (Original) The method as described in claim 14 wherein the set of application programming interfaces include login, commit, abort and logout APIs.

16. (Previously Presented) The method as described in claim 14 wherein the authentication framework is compliant with an authentication service of a virtual machine.

17. (Previously Presented) A computer program product in a computer readable medium for enabling a program written in untrusted code to access a native operating system resource, the computer program product comprising the steps of:

means for listening on a named pipe by a trusted login service for login requests;

means responsive to a login request for requesting a native operating system identifier by the trusted login service, wherein the login request contains an identifier for a uniquely-named response pipe,;

means for returning to the program via the uniquely-named response pipe the native operating system identifier, wherein the uniquely-named response pipe and the named pipe are not identical;

in an authentication framework, using the native operating system identifier to create a credential object; and

using the credential object to login to the native operating system to enable the program to access the resource.

18. (Previously Presented) The computer program product as described in claim 17 wherein the program executes in a virtual machine supported by the native operating system and the native operating system supports named-pipe servers.

19. (Previously Presented) The computer program product as described in claim 17 wherein the program is written in an interpreted language.

B1

20. (Previously Presented) The computer program product as described in claim 17 wherein the authentication framework is compliant with an authentication service of a virtual machine.

21. (Previously Presented) An application server, comprising:

a set of programs that are supported by a virtual machine that is supported by a native operating system;

a processor running the native operating system providing support for executing the set of programs; and

means for enabling each program in the set of programs to run in an operating system thread while impersonating a different native operating system user in accordance with a token that was created during a login operation in the native operating system and that was associated with a program while the program was acting as a named-pipe server to listen for a login response on a named pipe that was uniquely created for a login request to obtain the token, wherein the login request contained an identifier for the named pipe.

22. (Previously Presented) The application server as described in claim 21 wherein the native operating system supports named-pipe servers.

B1

23. (Previously Presented) The application server as described in claim 21 further including a server application executed by the processor for receiving a request for service from a client machine and initiating execution of a program in the set of programs in a given operating system thread.

24. (New) A method for enabling a program written in untrusted code to access in a trusted manner a resource supported on a computing device executing a native operating system, the method comprising:

listening, by a trusted login service in the native operating system, for login requests on a named request pipe;

generating a login request at the program, wherein the login request contains authentication information and an identifier for a named response pipe, wherein the named request pipe and the named response pipe are not identical;

in response to creating the named response pipe by the program, acting as a named-pipe server on the named response pipe by the program;

in response to receiving the login request on the named request pipe at the trusted login service from the program, performing a login operation with the authentication information by the trusted login service into the native operating system;

in response to performing the login operation, sending a login response on the named response pipe from the trusted login service to the program;

B1

in response to receiving the login response on the named response pipe at the program from the trusted login service, closing the named response pipe such that the named response pipe is uniquely associated with the login request and is not used for additional login requests;

in response to receiving the login response on the named response pipe at the program from the trusted login service, creating a credential object by the program using a token generated during the login operation; and

using the credential object by the program to access the resource within the native operating system.

25. (New) The method as described in claim 24 further comprising:

prior to sending the login response but after performing the login operation, performing a first impersonation operation by the trusted login service, wherein the first impersonation operation is based on the token such that the trusted login service impersonates a security context of a user associated with the authentication information; and

prior to closing the named response pipe but after receiving the login response, performing a second impersonation operation by the program, wherein the second impersonation operation is based on the trusted login service acting as a named-pipe client on the named response pipe and the program acting as the named-pipe server on the named response pipe such that the program impersonates a security context of the login response as a most recent message read from the named response pipe.

26. (New) The method as described in claim 25 further comprising:

B1

after sending the login response, performing a first revert operation by the trusted login service to terminate its previous impersonation; and

after performing a token duplication operation by the program, performing a second revert operation by the program to terminate its previous impersonation.

27. (New) The method as described in claim 25 further comprising:

performing a token duplication operation by the program, wherein the token duplication operation associates the token with a thread that initiated a login in order to obtain access to the resource.

28. (New) The method as described in claim 27 further comprising:

after performing the token duplication operation, performing a third impersonation operation by the program, wherein the third impersonation operation is based on the credential object such that a calling thread impersonates a security context of a user associated with the authentication information.